Please type a plus sign (+) inside this box [+]

PTO/SB/05 (12/97)
Approved for use through 09/30/00. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b)

Attorney Docket No. 42390.P7268 Total Pages 2

First Named Inventor or Application Identifier Fred Gruner

Express Mail Label No. EL431892167US

ADDRESS TO: **Assistant Commissioner for Patents**
Box Patent Application
Washington, D. C. 20231

APPLICATION ELEMENTS
See MPEP chapter 600 concerning utility patent application contents.

1. X Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)

2. X Specification (Total Pages 27)
(preferred arrangement set forth below)
- Descriptive Title of the Invention
- Cross References to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to Microfiche Appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings (if filed)
- Detailed Description
- Claims
- Abstract of the Disclosure

3. X Drawings(s) (35 USC 113) (Total Sheets 8)

4. X Oath or Declaration (Total Pages 5)

a. ___ Newly Executed (Original or Copy)

b. ___ Copy from a Prior Application (37 CFR 1.63(d))
(for Continuation/Divisional with Box 17 completed) **(Note Box 5 below)**

i. ___ DELETIONS OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).

5. ___ Incorporation By Reference (useable if Box 4b is checked)
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

6. ___ Microfiche Computer Program (Appendix)

7. ____ Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)
a. ____ Computer Readable Copy
b. ____ Paper Copy (identical to computer copy)
c. ____ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. ____ Assignment Papers (cover sheet & documents(s))

9. ____ a. 37 CFR 3.73(b) Statement (where there is an assignee)

__X__ b. Power of Attorney

10. ____ English Translation Document (if applicable)

11. ____ a. Information Disclosure Statement (IDS)/PTO-1449

____ b. Copies of IDS Citations

12. ____ Preliminary Amendment

13. __X__ Return Receipt Postcard (MPEP 503) (Should be specifically itemized)

14. ____ a. Small Entity Statement(s)

b. Statement filed in prior application, Status still proper and desired

15. ____ Certified Copy of Priority Document(s) (if foreign priority is claimed)

16. __X__ Other: Certificate of Express Mail with copy of postcard showing contents of Express Mail package.

17. **If a CONTINUING APPLICATION,** check appropriate box and supply the requisite information:

___ Continuation ____ Divisional ____ Continuation-in-part (CIP)

of prior application No: _____

18. **Correspondence Address**

____ Customer Number or Bar Code Label _____
(Insert Customer No. or Attach Bar Code Label here)

or

__X__ Correspondence Address ~~Below~~

NAME Ronald C. Card Ronald C. Card Aug. 30, 1999

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

ADDRESS 12400 Wilshire Boulevard

Seventh Floor

CITY Los Angeles STATE California ZIP CODE 90025-1026

Country U.S.A. TELEPHONE (408) 720-8598 FAX (408) 720-9397

PTO/SB/05 (12/97)
Approved for use through 09/30/00. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

042390.P7268

PATENT

UNITED STATES LETTERS PATENT APPLICATION

FOR

METHOD AND SYSTEM FOR A TWO STAGE PIPELINED INSTRUCTION DECODE AND ALIGNMENT

Inventors:

Fred Gruner
Mike Morrison
Kushagra Vaid

Prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Boulevard, 7th Floor
Los Angeles, California 90025-1026
(408) 720-8598

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EL431892167US

Date of Deposit 8/30/99

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner
of Patents and Trademarks, Washington, D.C. 20231.

JUANITA BRISCOE
(Typed or printed name of person mailing paper or fee)

Juanita Briscoe
(Signature of person mailing paper or fee)

METHOD AND SYSTEM FOR A TWO STAGE PIPELINED INSTRUCTION DECODE AND ALIGNMENT

FIELD OF THE INVENTION

The present invention pertains generally to the field of computer systems and more particularly to a system and method for decoding the links of macrocode instructions within pipelined or super pipelined microprocessors.

BACKGROUND OF THE INVENTION

Processors (including, but not limited to, general and special purpose microprocessors, micro-controllers, and digital signal processors (DSPs)) typically include execution units that execute a sequence of instructions, termed micro-instructions, derived from a computer program. Many computer programs are written in a high level language that is not directly executable by the central processing unit (CPU) of a computer and the instructions of such programs must accordingly be decoded into a form suitable for execution by the CPU. For example, a program may be written in a high level language such as C, C++, or Java, and then compiled into a corresponding sequence of macro-instructions, which are in turn decoded into micro-instructions for eventual execution. Programs can also be written directly of a series of macro-instructions (that is, machine code).

Macro-instructions are commonly stored as contiguous data blocks in a memory resource, such as main memory (ergo, RAM) or in a cache, for retrieval and supplied to a decoder unit within a processor for decoding into micro-instructions. To enable the decoder unit successfully to decode macro-instructions, it will be appreciated that it is necessary to identify instruction boundaries within retrieve data blocks, that constitute the instruction stream, that indicate where one macro-instruction ends and the next begins.

The task of identifying such instruction boundaries by processors having complex instruction set (CISC) architectures, such as the Intel architecture (IA) developed by Intel Corporation of Santa Clara, California, is complicated by the use of a variable-length instruction set (e.g., the Intel architecture (IA) instruction set). Specifically, in reduced instruction set computer (RISC) processor architectures and instruction sets, macro-instructions typically had a fixed length, in which case the boundaries between instructions can be determined with relative ease once an initial boundary is identified, as each instruction has a known length. For a variable-length instruction set, once an initial boundary location is identified, the length of each macro-instruction must be ascertained to identify subsequent instruction boundaries. The task of identifying boundaries is further complicated by a variable-length instruction set that, for the purposes of supporting legacy programs, supports multiple data and addressing sizes.

SUMMARY OF THE INVENTION

A system and method for aligning a macro instruction stream is described. In one embodiment, the system comprises a rotator logic unit for rotating data bytes of the macro instruction stream. A shifter logic unit is used for shifting the data bytes to the start of a macro instruction based upon a length of an immediately prior macro instruction. The rotator and shifter operated during a single clock cycle.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features and advantages of the present invention will be apparent to one skilled in the art in light of the following detailed description in which:

Figure 1 is a block diagram illustrating an exemplary format of a macro-instruction consisting of bytes that may be decoded according to the present invention.

Figure 2 is a block diagram showing an architecture of an exemplary microprocessor within which the present invention may be implemented.

Figure 3 is a block diagram showing architectural details regarding an instruction fetch engine and an instruction translate engine, according to exemplary embodiment of the present invention, that may be incorporated into the microprocessor shown in **Figure 2.**

Figure 4 is block diagram showing architectural details of one embodiment of an instruction length decoder (ILD) within the instruction pipe.

Figure 5 is a block diagram showing architectural details of a portion of the ILD.

Figure 6a is a block diagram of one embodiment of a portion of the macro-instruction shown in **Figure 1** consisting of bytes 0-3 that may be decoded according to the present invention

Figure 6b is a block diagram of one embodiment of an instruction length vector.

Figure 7 is a block diagram showing architectural details of one embodiment for the length decode unit.

Figure 8 is a flow diagram illustrating one embodiment of the instruction length alignment process.

DETAILED DESCRIPTION

A system and method for a two stage instruction length decode and alignment of macro-instructions are described.

In the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that the present invention may be practiced without these specific details. In some instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention.

Figure 1 is a diagrammatic representation of an exemplary macro-instruction 100 consisting of bytes that may be decoded according to the teachings of the present invention. Specifically, **Figure 1** illustrates the format of an exemplary macro-instruction forming the part of the Intel architecture (IA) instruction set, as developed by Intel Corporation of Santa Clara, California. For purposes of the present specification, the terms "macro-instruction" and "instruction" shall both be taken to refer to what is commonly understood to be a macro-instruction and not a micro-instruction.

As defined within the Intel architecture instruction set, an exemplary macro-instruction 100 may comprise zero to fourteen instruction prefixes 102 (each instruction prefix 102 being one byte in length), an opcode 104 (1-2 bytes in length), a ModR/M operand indicator 106 (0-1 byte in length), an SIB of 108 (0-1

lengths in byte), address displacement 110 (0-4 bytes in length), and an intermediate data constant 112 (0-4 bytes in length). Opcode 104 may be either one or two bytes in length. For two-byte opcodes, the first byte is 0F.

ModR/M, if present, is one byte in length and comprises a mod field 114, reg/opcode field 116 and the R/M field 118. The mod field 114 combines with the R/M field 118 to form 32 possible values: 8 register and 24 addressing modes. The reg/opcode field 116 specifies either a register number or three more bits of opcode information. The reg/opcode field 116 use is specified in the first byte of the primary opcode 104. The R/M field 118 may specify a register as operand or may be combined with the mod field 114 to encode an addressing mode. Certain encodings of the ModR/M byte 106 require a second addressing byte, the Scale Index Base (SIB) byte 108, to fully specify the addressing mode of the instruction. The base-plus-index and base-plus-scaled-index forms of 32-bit addressing require the SIB byte 108. SIB byte 108 includes a scale field 120, which specifies the scale factor, index field 122, which specifies the register number of the index register, and base field 124 which specifies the register number of the base register.

Current processors, such as for example the Pentium Pro® or the Pentium II® Processors, may operate in either a 16- or 32-bit mode. Each macro-instruction may be decoded and executed as:

1. A fixed 8-bit data/address instruction;
2. A fixed 16-bit data/address instruction; or

3. A variable 16- or 32-bit data/address instruction, as determined by a mode operation bit (commonly termed a D-bit) within the IA processor.

Figure 2 is a block diagram showing an exemplary microprocessor 30 within which the present invention may be implemented. The microprocessor 30 is pipelined and includes in-order front-end circuitry 32 and out-of-order back-end circuitry 34. The out-of-order back-end circuitry 34 executes micro-instructions in an out-of-order fashion and retires executed micro-instructions in an in-order fashion according to the original sequence in the macro instruction. The front-end circuitry 32 comprises an instruction fetch engine 36 that retrieves macro-instructions, which may conform to the format illustrated in **Figure 1**, via a bus interface unit 37 from a main memory (not shown) associated with a microprocessor 30, or from an internal unified cache 48 that caches both macro-instructions and data. In an alternative embodiment, cache 48 may be located downstream of an instruction translate engine 38 and may cache decoded micro-instructions derived from macro-instructions. Macro-instructions retrieved by the instruction fetch engine 36 are then propagated to the instruction translate engine 38 that translates macro-instructions into corresponding micro-instructions. Micro-instructions are issued from the instruction translate engine 38 to a control unit 40 (also referred to as a microcode unit), that forms part of the back-end circuitry 34, and includes a microcode sequencer (MS) 41 and a

microcode control read-only memory (ROM) 43. The control unit 40 interprets the micro-instructions sent to it, and handles exceptions, break points, and interrupts. From the control unit 40, micro-instructions are dispatched to a pipeline including an address generation unit 42, an integer execution unit 44 (also known as an arithmatic/logic unit (ALU)) and/or a floating point execution unit 46.

The microprocessor 30 further includes a page unit 50 that translates linear addresses into physical addresses, and includes at least one translation look-aside buffer (TLB) for this purpose.

Figure 3 is a block diagram showing further details regarding the instruction fetch engine 36 and the instruction translate engine 38 of the exemplary microprocessor shown in **Figure 2**. The instruction fetch engine 36 is shown to include a macrocode instruction queue (MIQ) 302 for receiving macro-instructions from main memory 47 or unified cache 48. Alignment buffers 304 buffer macro instructions before they are dispatched to the instruction translate engine 38.

The alignment unit (ALN) 306 is responsible for aligning the instruction stream. ALN 306 determines where a fetch set of data the instruction to be executed begins and ends. A pointer within alignment buffers 304 is adjusted to point to the next instruction to be executed in the processor. Using the length of the previous instruction and the known starting point of the previous

instruction, ALN 306 shifts the data stream pointer in order to align the pointer to the beginning of the next instruction in the data stream. ALN 306 aligns the instruction one pipe stage behind the length decode of the instruction. The present invention decodes one instruction per cycle within two pipe stages.

In the second pipe stage, instruction length decoder (ILD) 308 determines the length of the current instruction. IA instructions are variable length instructions varying in length from 1 to 15 bytes with prefixes and 1 to 11 bytes without prefixes. In order to properly align and decode the instructions, the length of the instruction must be determined. The bytes that are received from the ALN 306 stage are assumed to start with the first byte of instruction. The ILD 308 decodes these instruction bytes, determines the length of the instruction, and sends the length to the ALN 306 for subsequent instruction realignment and to the DE1 stage for marking the instruction boundaries.

ILD 308 decodes instruction lengths in one pipe stage. All instruction lengths are computed in the same clock cycle. Instructions with prefixes take up to N+1 clock cycles, where N is the number of prefixes preceding the instruction. In one embodiment, the parallel computation of the instruction data is used to determine the lengths of individual components and then these components are merged together to determine the total resulting length. Decoding length information in one cycle and using this length information for alignment in the same cycle allows the present invention to reduce overhead performance loss. In

addition, in one embodiment, less chip area may be utilized in order to implement the present invention.

ILD 308 creates an output length vector, which is utilized in decoder 310 for the decoding of the instruction. In one embodiment, a two-stage alignment-decode mechanism for the alignment (ALN) and length (LEN) or decode stages is used. ALN 306 includes one or more buffers for holding fetched information, and for communication with the fetch engine 36 for fetching instructions from memory or cache. ALN 306 precedes ILD 308 where instructions are initially decoded. ALN 306 fetches one or more instructions from memory or cache and determines where the next instruction begins and ends. Instruction data is fetched from the cache and stored in the buffer, or in one of the plurality of buffers. ILD 308 determines the length of the instruction in order to align the next instruction within the buffer.

Figure 4 is a block diagram showing architectural details of one embodiment of an instruction length ALN 306 and ILD 308 with the instruction pipe. Length Decode (LEN) 402 is housed within the ILD 308. In addition, ILD 308 contains back-end circuitry 404 for processing the decoded instructions.

LEN 402 receives the instruction data from ALN 306 and determines the length of the instruction. The instruction length is used by the ALN 306 to shift the next instruction. In addition, LEN 402 calculates the prefix count and shifts the instruction lengths by the prefix count. If the length of the instruction

exceeds 15 bytes, LEN 402 flags a length violation. The decoded instructions are passed to back-end circuitry 404 for processing.

LEN 402 is also responsible for decoding prefixes. Prefix State Machine looks at instruction byte 0 to determine if a prefix was encountered and to determine the prefix type. Prefix State Machine also counts prefixes it has encountered for a particular instruction.

Figure 5 is a block diagram showing architecture details of ALN 306 and LEN 402. Data stream bytes are received into two 10x16-byte buffers 306 from MIQ buffers 302. The data stream is rotated into rotator 504. In one embodiment, rotator 504 consists of 12 bytes. Rotator 504 rotates the data bytes of two instructions. In one embodiment, an instruction has a maximum length of 11 bytes (without prefixes). If 12 bytes captures both instructions, then ALN 306 will have a maximum throughput. Rotator 504 is one pipe stage behind the decoding of the current instruction. Using the length vector obtained from the LEN 402, ALN 306 next shifts the current instruction into shifter 506. Shifter 506 shifts to the exact instruction start based on the length of the first instruction in the data stream. If rotator 504 does not contain the entire instruction required by shifter 506, rotator 504 rotates instruction data from buffers 304. Shifter 506 output gives the current instruction for the current pipe stage. It is assumed that the first instruction within the data stream begins at the beginning of the data buffer. Thus, during the current pipe stage, rotator 504 is obtaining instruction

data for the current instruction while shifter 506 is obtaining data for the next instruction in the data stream.

Referring to Table 1, an example of the alignment of one embodiment is shown. Table 1 shows the contents of buffers 304. In this example, it is assumed that instruction A is being executed and has a length of 5 bytes. This example assumes that no branches are present in the current buffer.

Table 1

	Buffer 1
Byte 1	
Byte 2	A
Byte 3	
Byte 4	
Byte 5	
Byte 6	
Byte 7	B
Byte 8	
Byte 9	
Byte 10	C
Byte 11	
Byte 12	

Initially, rotator 504 contains bytes 2-13 as shown in Table 2 for time 1. The shifter 506 contains bytes 2-13 and length is 0. LEN 402 determines the length of A. The length of A is returned to shifter 506. At time 2, shifter 506, using the length of A of 5 bytes, shifts bytes from rotator 504 into shifter 506 offset by the length of A and shifts A to LEN 402. Thus, rotator 504 contains bytes 2-13, the shifter 506 now contains bytes 7-13, and LEN 402 contains bytes 2-6 (instruction A). Shifter 506 then shifts bytes 7-13 to LEN 402 for length determination of instruction B. At time 3, shifter 506, using the length of B of 3 bytes, shifts instruction B into LEN 402, bytes 10-18 are shifted into shifter 506, and rotator contains bytes 7-18. The process is repeated in order to shift instruction C into LEN 402 as shown in Table 2.

Table 2

Time	1	2	3	4
ALN Rotator Output (Buffer Bytes)	2-13	2-13	7-18	10-21
ALN Shifter Output (Buffer Bytes)	2-13	7-13	10-18	12-21
Length in LEN	0	5	3	2
Instruction in LEN		A	B	C

Instruction prefixes 102 are handled as length 1 instructions. This enables the LEN 308 to decode the prefixes one prefix at a time. When the two instruction buffers 304 do not represent consecutive addresses (that is, a branch is predicted to be taken), a clock cycle is used to clean-up and reset the read pointer to the target of the branch.

Referring to **Figure 6a**, the first four bytes of a macro-instruction are shown in two configurations. Configuration 1 (710) shows a one-byte opcode 104, the ModR/M byte 106, and the SIB byte 108. The opcode 104 is in byte 0 (702), the ModR/M byte 106 is in byte B1 (704), and the SIB byte 108 is in byte B2 (706). The second configuration 720 shows a two-byte opcode 104 configuration. In this configuration, the opcode 104 occupies bytes B0 (702) and B1 (704), the ModR/M byte 106 occupies byte B2 (706), and the SIB byte 108 occupies byte B3 (708).

Figure 7 is a block diagram showing architectural details of one embodiment of the instruction length decoder (LD) 508. LD 508 determines the length of various portions of the instruction received from ALN 306. ALN 306 shifts the current instruction from the shifter 506 onto the LD 508. Within the LD 508, opcode-plus-immediate logic unit 602 determines the length of the opcode 104 and immediate data 112 of the current instruction. B0 and B1 are inputs to OPIMM 602 together with the operand-size (Osz) signal. The Osz signal selects the sizes of operands that instructions operate on. When the 16-bit Osz signal is in force, operands may be either 8 or 16 bits. When the 32-bit Osz signal is in force, operands may be 8 or 32 bits.

Simultaneously, a ModR/M present logic unit (MODBRNT) 604 determines if the ModR/M 106 is required. B0 and B1 are input into the MODPRNT 604 logical unit. The MODPRNT 604 logical unit performs a lookup of the opcode to determine whether a ModR/M byte is required.

LD 508 also determines memory address displacement length 614 for address displacement 110. LD 508 assumes that the ModR/M byte 106 is present and determines memory address displacement length 614 for both the one-byte and two-byte opcodes. A memory-length-one logic unit (MEMLEN1) 606 determines an anticipatory length of the memory displacement based on an assumption that a one-byte opcode 104 is present. MEMLEN1 606 uses the input from bytes B1 and B2. This corresponds to configuration 710 of **Figure 6a**. In addition, an address-size (Asz) signal is provided to MEMLEN1 606. The Asz

signal selects the sizes of addresses used to address memory, either 16 or 32 bits. When the 16-bit Asz signal is in force, segment offsets and displacements are 16-bits. This limits the size of a segment that may be addressed to 64 KiloBytes. When the 32-bit Asz signal is in force, segment offsets and displacements are 32-bits, allowing segments up to 4 GigaBytes to be addressed.

A memory-length-two logic unit (MEMLEN2) 608 determines an anticipatory length of the memory based on an assumption that a two-byte opcode 104 is present. The MEMLEN2 608 uses the input from B2 and B3, corresponding to the second configuration 720 of **Figure 6a**, together with Asz. The output from MEMLEN1 606 and MEMLEN2 608 are multiplexed by multiplexer (MUX) 610, and a MEMLEN is output depending upon whether a one-byte or two-byte opcode is present.

The outputs from OPIMM 602, MODBRNT 604, and from MUX 610 are combined together by a length-summation logic unit 612. Length-summation logic unit 612 adds the lengths based on whether the ModR/M byte 106 is required by the opcode 104. Length-summation logic unit 612 creates two outputs: an instruction valid signal and an instruction length vector.

Using an instruction valid range input, length-summation logic unit 612 determines if the instruction is a valid instruction by combining the instruction valid range and the instruction length. In one embodiment, an instruction may be from 1 to 11 bytes in length.

The instruction length vector is an 11-byte vector as shown in **Figure 6b**. The instruction length vector is set to all zeroes except for the byte that indicates the length of the instruction, which is set to 1 for the length. Thus, if the instruction is 11 bytes in length, Len 11 is set to 1 and all other bits are set to 0. If the instruction is invalid, the instruction length vector is set to all zeroes. If the instruction is invalid, the shifter 506 does not shift the next instruction into the length decode and a clock cycle is required to realign the instructions.

Table 3 shows the possible outputs from OPIMM 602. The outputs are dependent on whether the opcode 104 is one or two bytes and the possible lengths of the immediate data 112. The immediate data may be 1, 2, 4 or 6 bytes in length. Thus, the opcode plus immediate may be 1, 2, 3, 4, 5, 6 or 7 bytes in length. Table 3 indicates the possible combinations of opcode and immediate displacement.

Table 3

OPCODE BYTE 1	OPCODE BYTE 2	1 BYTE IMM	2 BYTE IMM	4 BYTE IMM	6 BYTE IMM	OPCODE + IMM LENGTH
X	-	-	-	-	-	1
X	X	-	-	-	-	2
X	-	X	-	-	-	2
X	X	X	-	-	-	3
X	-	-	X	-	-	3
X	-	X	X	-	-	4
X	X	-	X	-	-	4
X	-	-	-	X	-	5
X	X	-	-	X	-	6
X	-	-	-	-	X	7

Table 4 indicates the possible outputs from MUX 610. Depending on whether a ModR/M or SIB byte are present and the valid, possible displacements, the displacement length is determined. The possible memory displacements are 1, 2 or 4 bytes. The possible memory displacements may be 1, 2, 3, 5 or 6 bytes in length. With the valid, possible combinations, it is not possible to have a memory displacement of four bytes in length.

Table 4

ModR/M PRESENT	SIB PRESENT	1 BYTE DISP	2 BYTE DISP	4 BYTE DISP	DISP LENGTH
Y	N	-	-	-	1
Y	Y	-	-	-	2
Y	N	X	-	-	2
Y	Y	X	-	-	3
Y	N	-	X	-	3
-	-	-	-	-	4 NOT POSSIBLE
Y	N	-	-	X	5
Y	Y	-	-	X	6

Table 5 shows the possible outputs for the instruction length vector of length summation logic unit 612. The table shows the combinations of the outputs from the OPIMM 602, MODPRNT 604, and MUX 610. Total instruction length may be from 1 to 11 bytes as indicated in Table 5.

Table 5

MEMLEN	OPIMM7	OPIMM6	OPIMM5	OPIMM4	OPIMM3	OPIMM2	OPIMM1	LENGTH
6	-	-	YES	-	-	-	-	11
6	-	-	-	NO	-	-	-	10
6	-	-	-	-	YES	-	-	9
6	-	-	-	-	-	YES	-	8
6	-	-	-	-	-	-	YES	7
5	-	NO	-	-	-	-	-	11
5	-	-	YES	-	-	-	-	10
5	-	-	-	NO	-	-	-	9
5	-	-	-	-	YES	-	-	8
5	-	-	-	-	-	YES	-	7
5	-	-	-	-	-	-	YES	6
4	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Not Applicable	-
3	NO	-	-	-	-	-	-	10
3	-	NO	-	-	-	-	-	9
3	-	-	YES	-	-	-	-	8
3	-	-	-	NO	-	-	-	7
3	-	-	-	-	YES	-	-	6
3	-	-	-	-	-	YES	-	5
3	-	-	-	-	-	-	YES	4
2	NO	-	-	-	-	-	-	9
2	-	NO	-	-	-	-	-	8
2	-	-	YES	-	-	-	-	7
2	-	-	-	NO	-	-	-	6
2	-	-	-	-	YES	-	-	5
2	-	-	-	-	-	YES	-	4
2	-	-	-	-	-	-	YES	3
1	NO	-	-	-	-	-	-	8
1	-	NO	-	-	-	-	-	7
1	-	-	YES	-	-	-	-	6
1	-	-	-	NO	-	-	-	5
1	-	-	-	-	YES	-	-	4
1	-	-	-	-	-	YES	-	3
1	-	-	-	-	-	-	YES	2
NO ModR/M	YES	-	-	-	-	-	-	7
NO ModR/M	-	YES	-	-	-	-	-	6
NO ModR/M	-	-	YES	-	-	-	-	5
NO ModR/M	-	-	-	YES	-	-	-	4
NO ModR/M	-	-	-	-	YES	-	-	3
NO ModR/M	-	-	-	-	-	YES	-	2
NO ModR/M	-	-	-	-	-	-	YES	1

Figure 8 is a flow diagram illustrating one embodiment of the instruction length alignment process. At block 802, ALN 306 stores instruction cache lines in 16-byte buffers 304. At block 804, ALN 306 rotates 12 bytes of the data stream

into rotator 504. Rotator 504 rotates the data bytes of two instructions. In one embodiment, an instruction has a maximum length of 11 bytes.

At block 806, ALN 306 uses the length vector obtains from LEN 402 to shift the current instruction into shifter 506. Shifter 506 shifts to the exact instruction start based on the length of the first instruction in the data stream. If rotator 504 does not contain the entire instruction required by shifter 506, rotator 504 rotates instruction data from buffers 502.

At block 808, ALN 306 outputs the current instruction to LEN 402 during the current pipe stage. In one embodiment, the first instruction within the data stream begins at the start of the data buffer. Thus, during the current pipe stage, rotator 504 is obtaining instruction data for the next instruction while shifter 506 is obtaining data for the current instruction in the data stream.

Several variations in the implementation for a system and method for two stage instruction length decode and alignment of macro-instructions have been described.

The specific arrangements and methods herein are merely illustrative of the principles of this invention. Numerous modifications in form and detail may be made by those skilled in the art without departing from the true spirit and scope of the invention.

CLAIMS

What is claimed is:

1. A method for aligning a macro instruction stream comprising:

rotating data bytes of the macro instruction stream; and

shifting the data bytes to the start of a macro instruction based upon a length of an immediately prior macro instruction.

2. The method of claim 1 wherein the rotating and shifting are performed during a single clock cycle.

3. The method of claim 1 further comprising:

receiving a length of an immediately prior macro instruction from a length decode logic unit.

4. The method of claim 1 further comprising:

storing macro instruction stream cache lines in alignment buffers prior to rotating the macro instruction stream.

5. The method of claim 1 wherein said shifting shifts to an exact start of the macro instruction.

6. The method of claim 1 further comprising:

providing the output of the shifting step to a length decode logic unit.

7. Logic for aligning a macro instruction stream comprising:

a rotator logic unit for rotating data bytes of the macro instruction stream;

a shifter logic unit for shifting the data bytes to the start of a macro instruction based upon a length of an immediately prior macro instruction.

8. The logic of claim 7 wherein the rotating and shifting are performed during a single clock cycle.

9. The logic of claim 7 further comprising:

a length vector for providing the length of an immediately prior macro instruction.

10. The logic of claim 7 further comprising:

alignment buffers for storing macro instruction stream cache lines for use by the rotator logic unit.

11. A processor to align a macro instruction stream comprising:

a rotator logic unit for rotating data bytes of the macro instruction stream;

a shifter logic unit for shifting the data bytes to the start of a macro instruction based upon a length of an immediately prior macro instruction.

12. The processor of claim 11 further comprising:

a length vector for providing the length of an immediately prior macro instruction.

13. A system for aligning a macro instruction stream comprising:

means for rotating data bytes of the macro instruction stream; and

means for shifting the data bytes to the start of a macro instruction based upon a length of an immediately prior macro instruction.

ABSTRACT

A system and method for aligning a macro instruction stream is described. In one embodiment, the system comprises a rotator logic unit for rotating data bytes of the macro instruction stream. A shifter logic unit is used for shifting the data bytes to the start of a macro instruction based upon a length of an immediately prior macro instruction. The rotator and shifter operated during a single clock cycle.

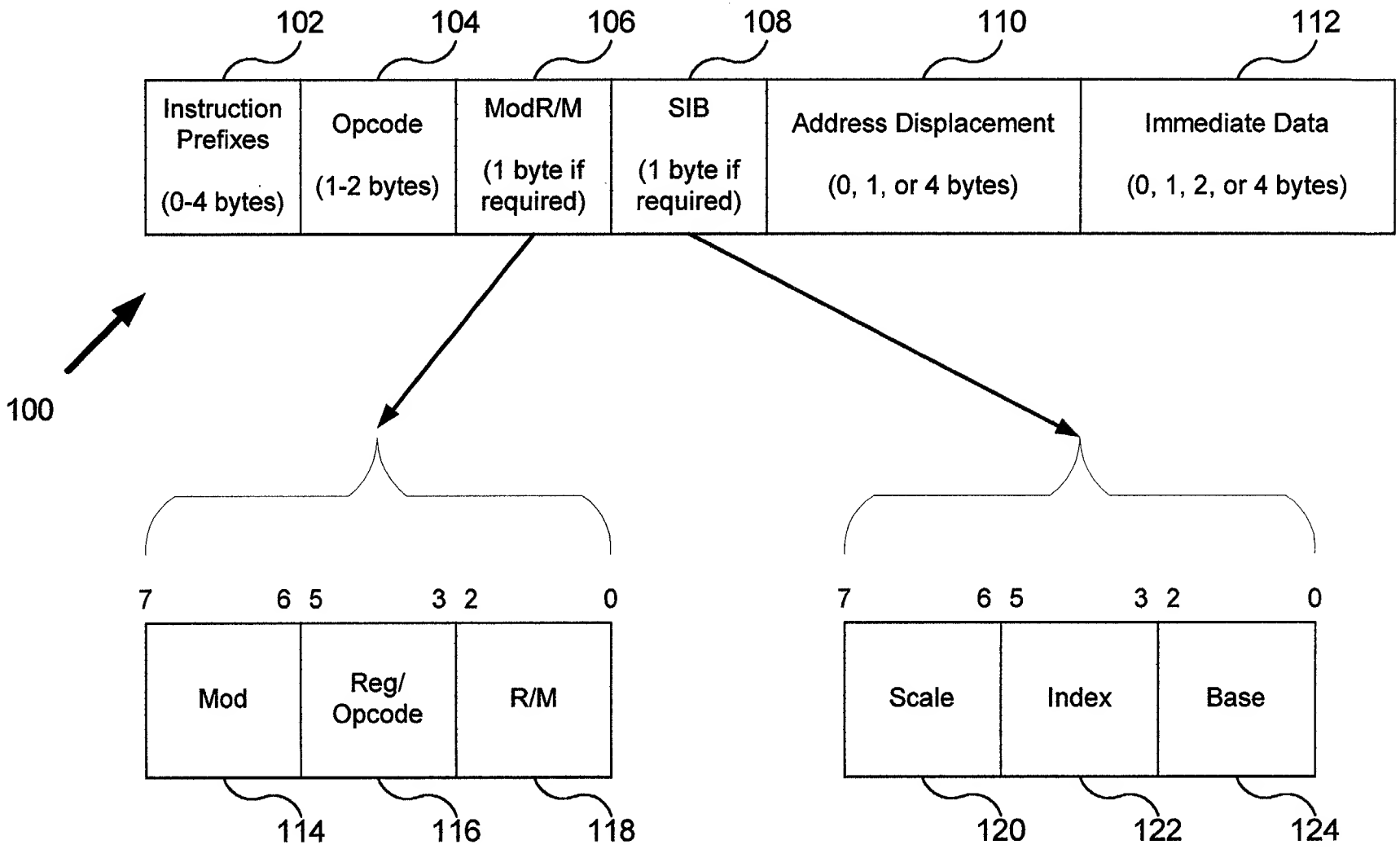
102
104
106
108
110
112
Instruction Prefixes (0-4 bytes)
Opcode (1-2 bytes)
ModR/M (1 byte if required)
SIB (1 byte if required)
Address Displacement (0, 1, or 4 bytes)
Immediate Data (0, 1, 2, or 4 bytes)
100
7 6 5 3 2 0
Mod
Reg/ Opcode
R/M
114
116
118
7 6 5 3 2 0
Scale
Index
Base
120
122
124

FIG. 1

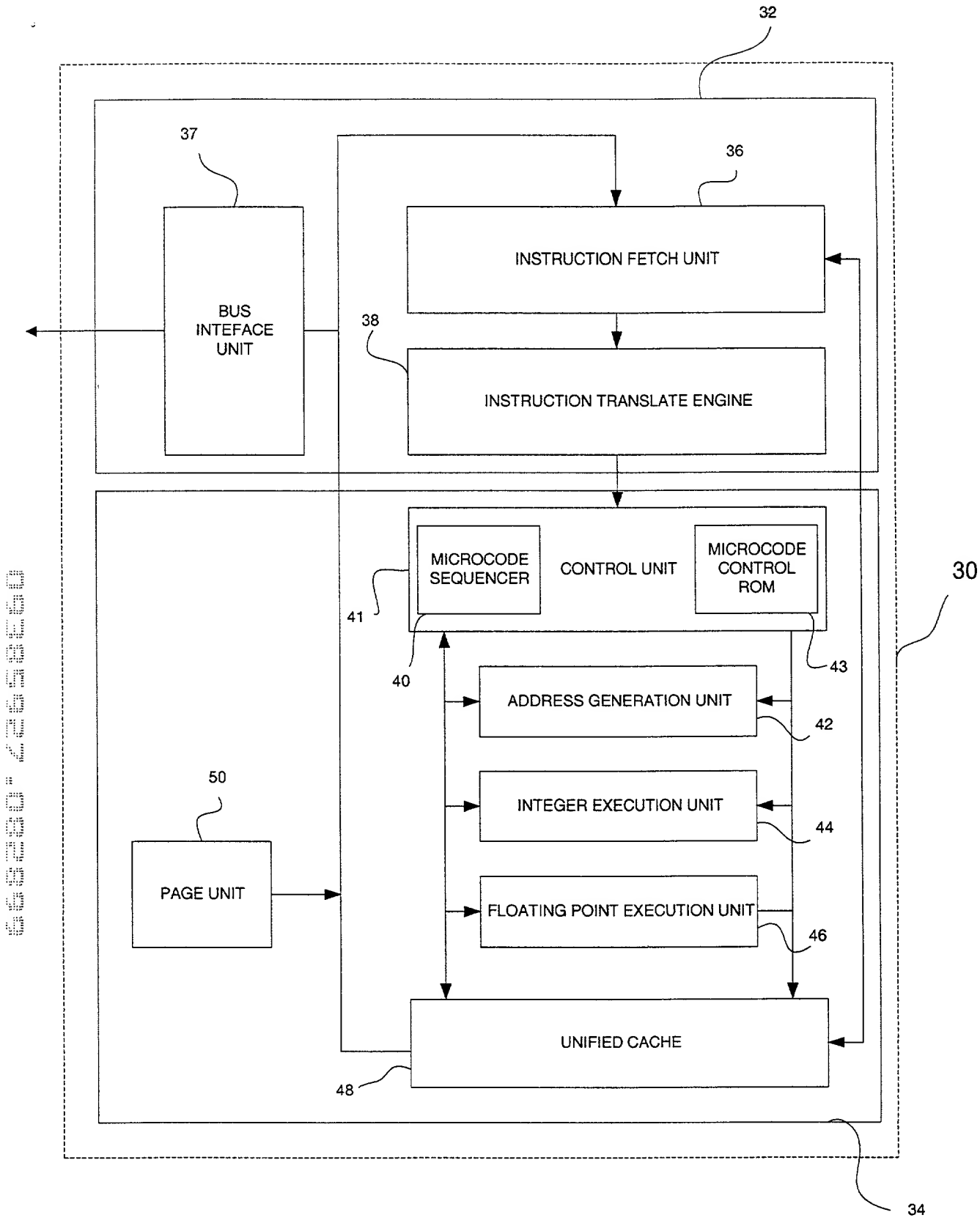
32
37
36
BUS
INTEFACE
UNIT
INSTRUCTION FETCH UNIT
38
INSTRUCTION TRANSLATE ENGINE
MICROCODE
SEQUENCER
CONTROL UNIT
MICROCODE
CONTROL
ROM
30
41
40
43
ADDRESS GENERATION UNIT
42
50
INTEGER EXECUTION UNIT
44
PAGE UNIT
FLOATING POINT EXECUTION UNIT
46
UNIFIED CACHE
48
34

FIGURE 2

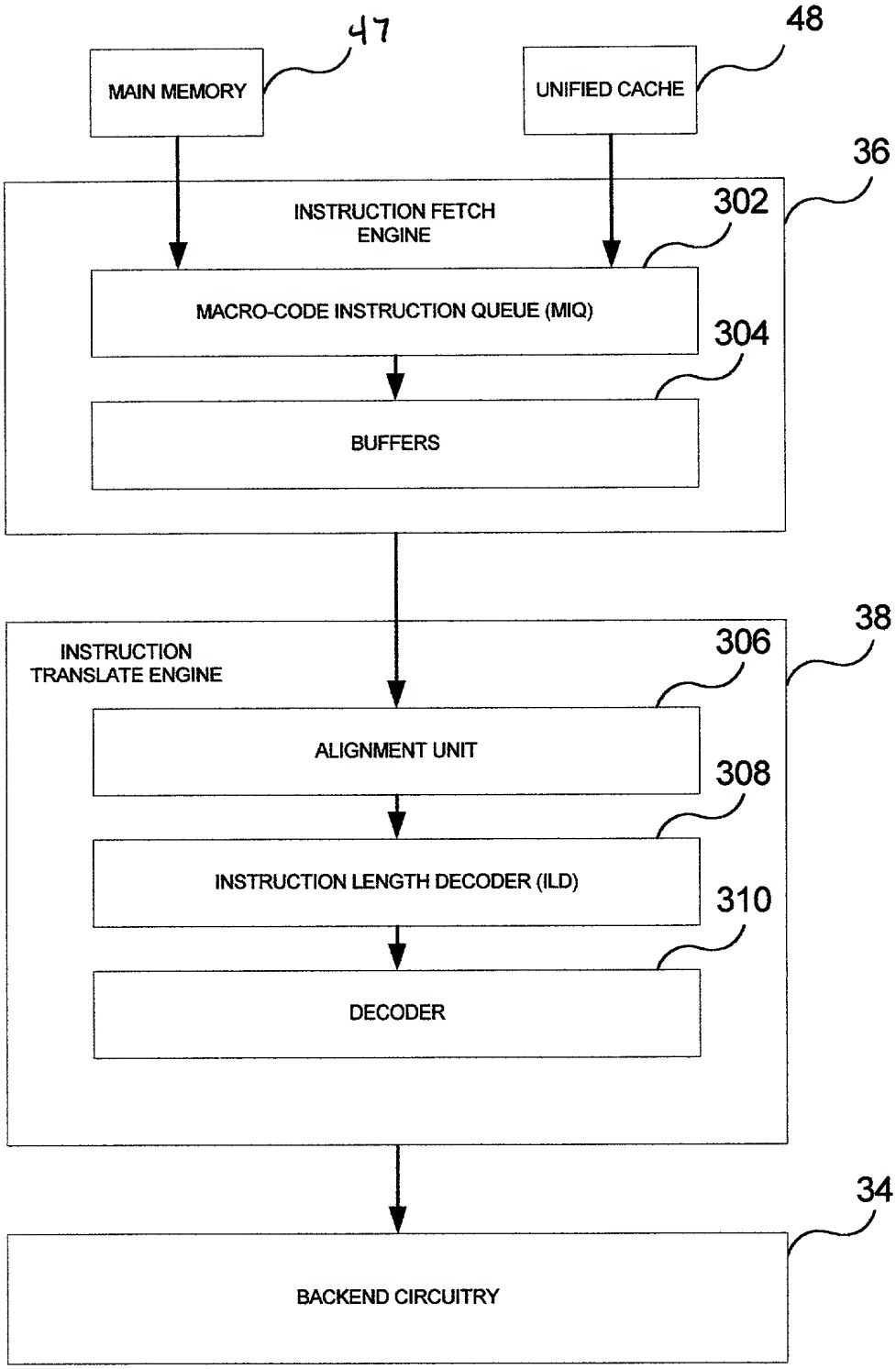
47
48
MAIN MEMORY
UNIFIED CACHE
36
INSTRUCTION FETCH ENGINE
302
MACRO-CODE INSTRUCTION QUEUE (MIQ)
304
BUFFERS
38
INSTRUCTION TRANSLATE ENGINE
306
ALIGNMENT UNIT
308
INSTRUCTION LENGTH DECODER (ILD)
310
DECODER
34
BACKEND CIRCUITRY

FIG. 3

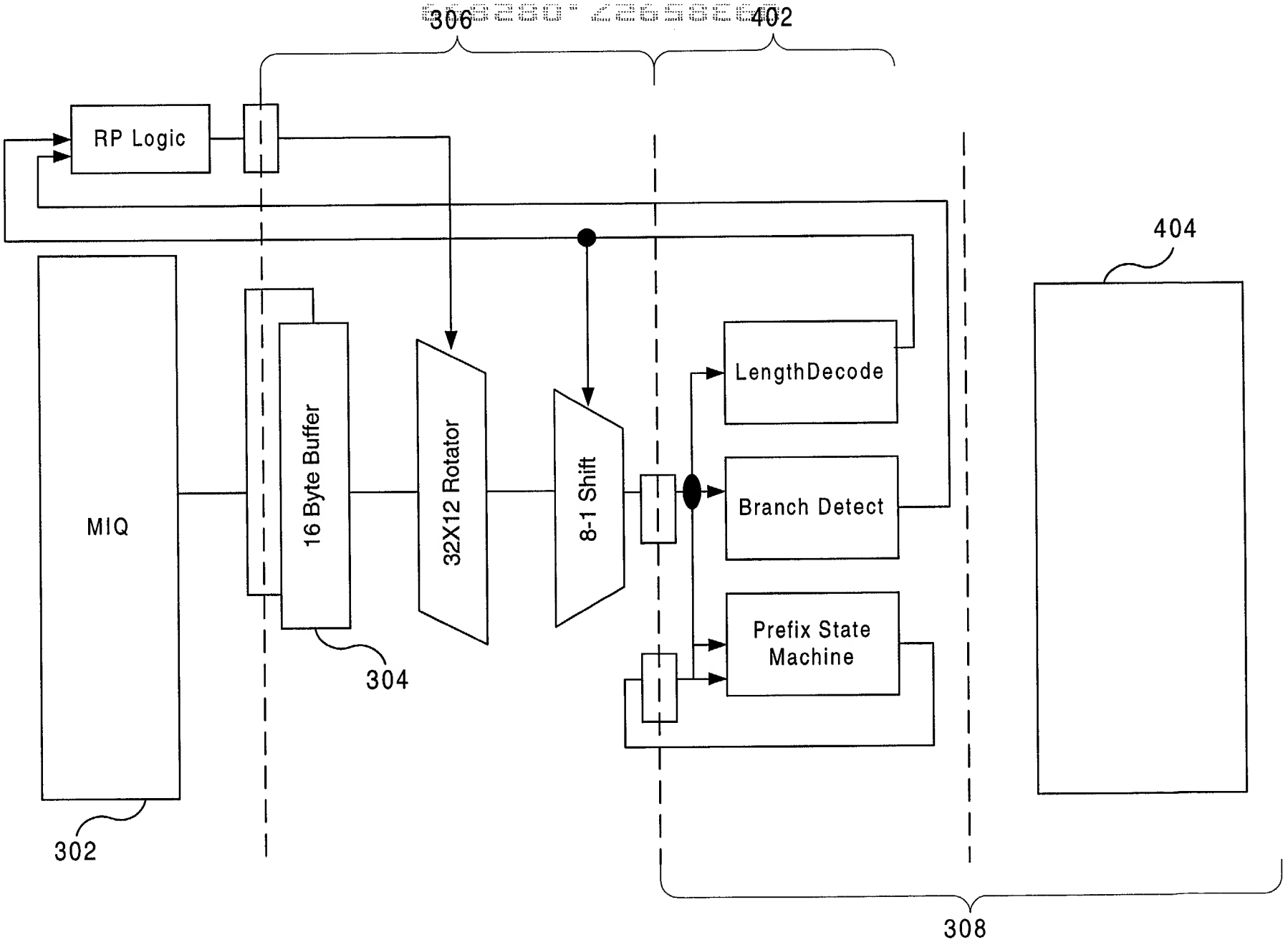
306
402
RP Logic
404
MIQ
16 Byte Buffer
32X12 Rotator
8-1 Shift
LengthDecode
Branch Detect
Prefix State Machine
304
302
308

FIG. 4

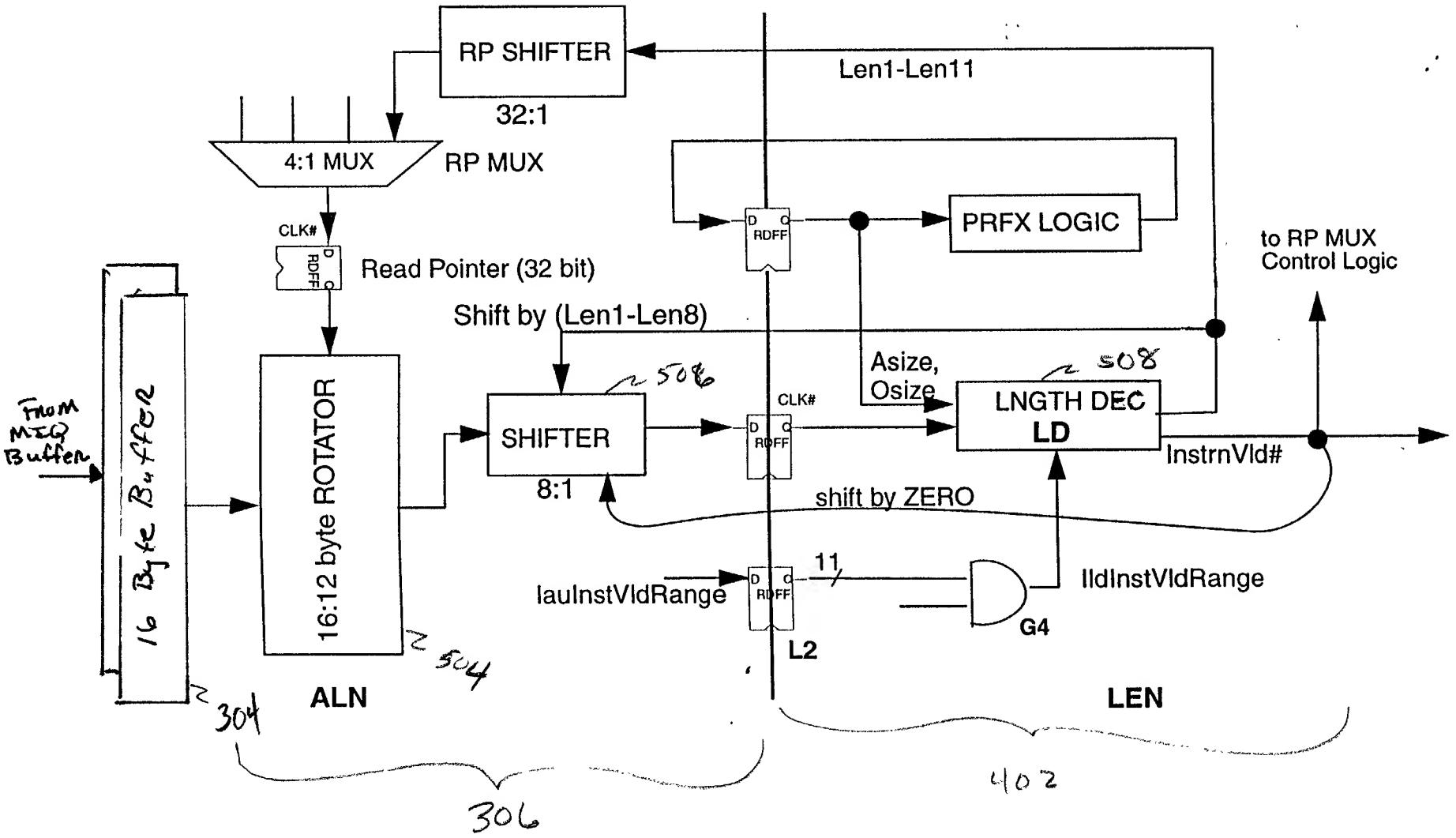

FIG. 5

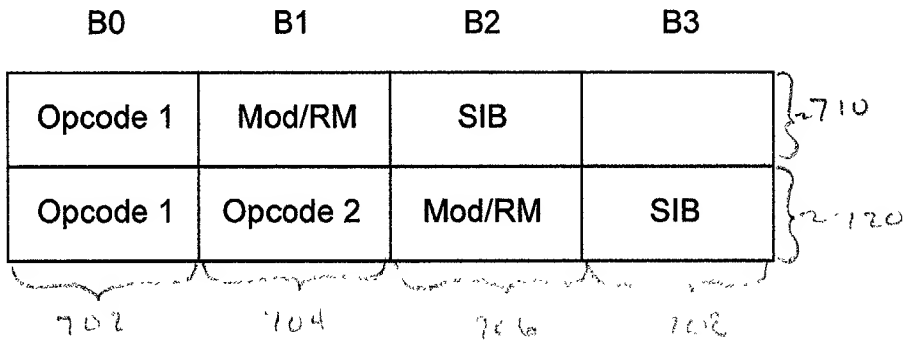

FIG. 6a

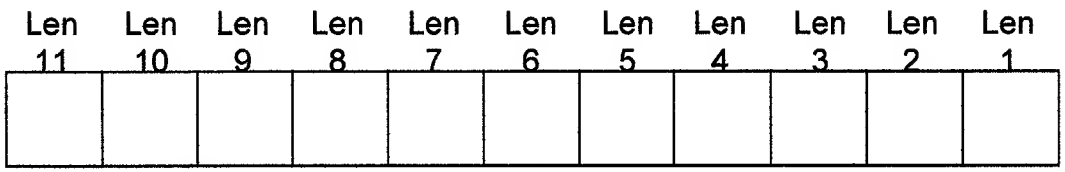

FIG. 6b

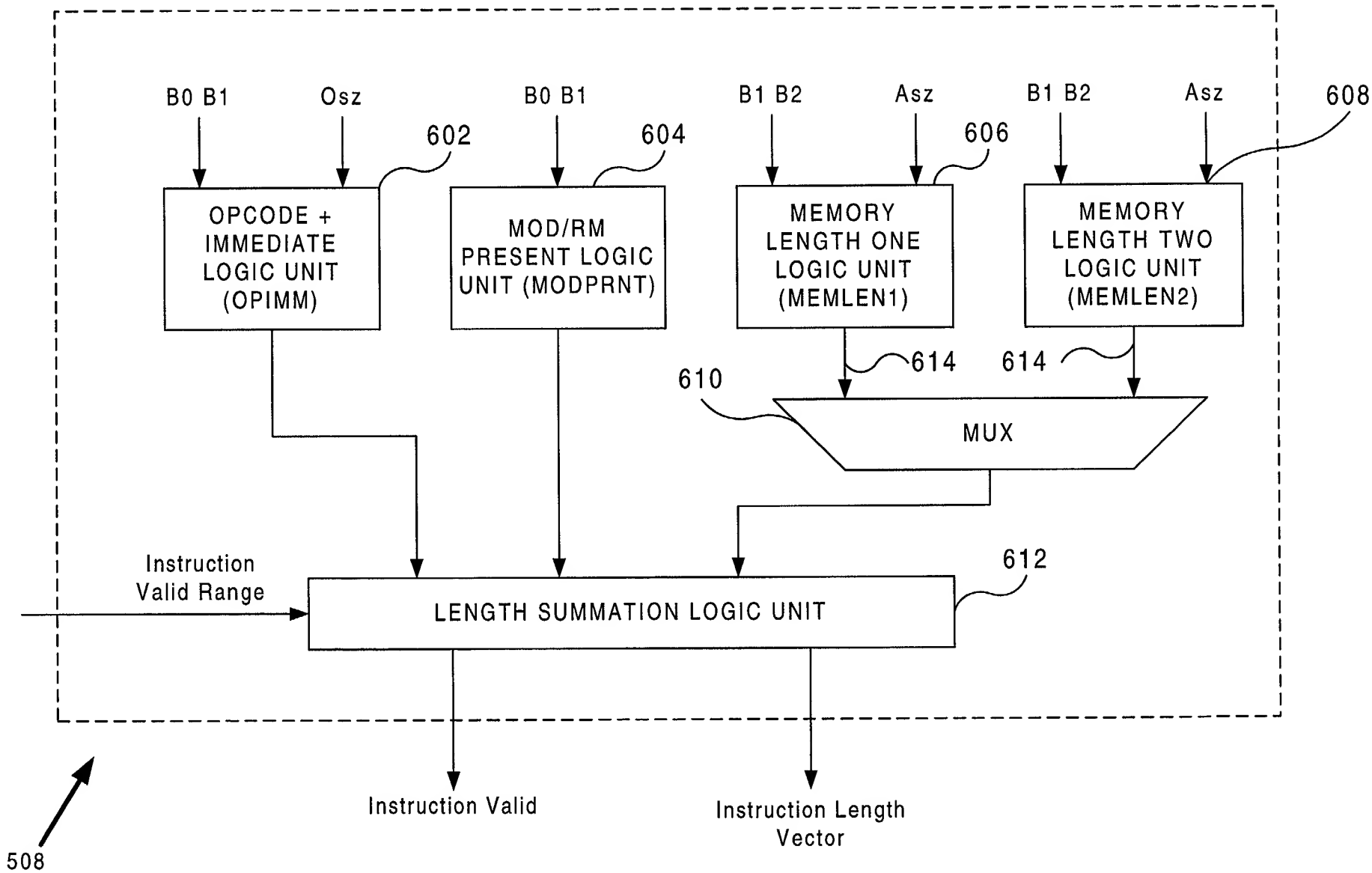

FIG. 7

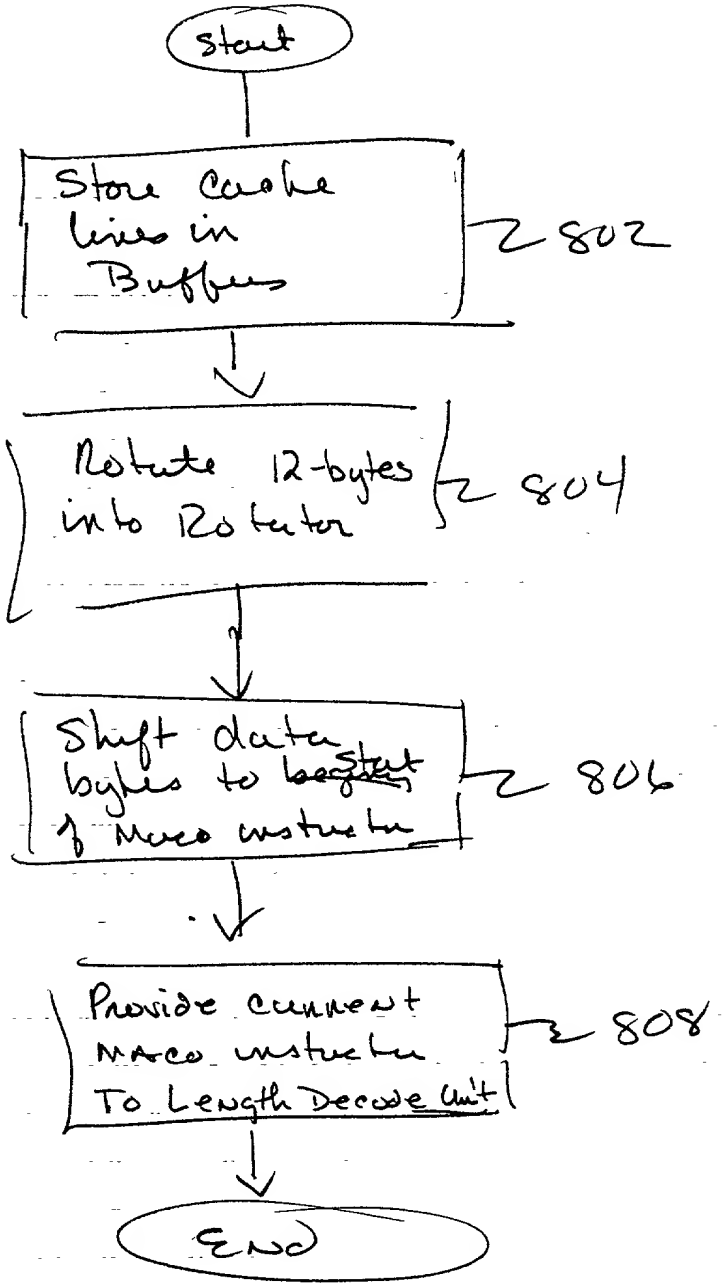
Start
Store cache lines in Buffers
802
Rotate 12-bytes into Rotator
804
Shift data bytes to start of MACC instruction
806
Provide current MACC instruction To Length Decode Unit
808
END

FIG. 8

P7268

Attorney's Docket No.: 42390.P7268 PATENT

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION
(FOR **INTEL CORPORATION** PATENT APPLICATIONS)

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

METHOD AND SYSTEM FOR A TWO STAGE PIPELINED INSTRUCTION DECODE AND ALIGNMENT

the specification of which

_X_ is attached hereto.
___ was filed on _____ as
United States Application Number _____
or PCT International Application Number_____
and was amended on _____.
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above. I do not know and do not believe that the claimed invention was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (for a utility patent application) or six months (for a design patent application) prior to this application.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d), of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below

Application Number	Filing Date
Application Number	Filing Date

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

Application Number	Filing Date	Status -- patented, pending, abandoned
Application Number	Filing Date	Status -- patented, pending, abandoned

I hereby appoint the persons listed on Appendix A hereto (which is incorporated by reference and a part of this document) as my respective patent attorneys and patent agents, with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

Send correspondence to Ronald C. Card **(Name of Attorney or Agent)**, **BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, 12400 Wilshire Boulevard 7th Floor, Los Angeles, California 90025 and direct telephone calls to** Ronald C. Card **(Name of Attorney or Agent)**, **(408) 720-8598.**

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor Fred Gruner

Inventor's Signature _____ Date _____

Residence Palo Alto, California (City, State) Citizenship United States of America (Country)

Post Office Address 2069 Edgewood Drive
Palo Alto, California 94303

Full Name of Second/Joint Inventor Mike Morrison

Inventor's Signature _____ Date _____

Residence Santa Clara, California (City, State) Citizenship United States of America (Country)

Post Office Address 3131 Homestead Road #10-J
Santa Clara, California 95051

Full Name of Third/Joint Inventor Kushagra Vaid

Inventor's Signature _____ Date _____

Residence Sunnyvale, California (City, State) Citizenship India (Country)

Post Office Address 988 Henderson Avenue, Apt. 4
Sunnyvale, California 94086

APPENDIX A

William E. Alford, Reg. No. 37,764; Farzad E. Amini, Reg. No. P42,261; Aloysius T. C. AuYeung, Reg. No. 35,432; William Thomas Babbitt, Reg. No. 39,591; Carol F. Barry, Reg. No. 41,600; Jordan Michael Becker, Reg. No. 39,602; Bradley J. Bereznak, Reg. No. 33,474; Michael A. Bernadicou, Reg. No. 35,934; Roger W. Blakely, Jr., Reg. No. 25,831; Gregory D. Caldwell, Reg. No. 39,926; Ronald C. Card, Reg. No. P44,587; Yong S. Choi, Reg. No. P43,324; Thomas M. Coester, Reg. No. 39,637; Michael Anthony DeSanctis, Reg. No. 39,957; Daniel M. De Vos, Reg. No. 37,813; Robert Andrew Diehl, Reg. No. 40,992; Tarek N. Fahmi, Reg. No. 41,402; James Y. Go, Reg. No. 40,621; Dinu Gruia, Reg. No. P42,996; Willmore F. Holbrow III, Reg. No. P41,845; George W Hoover II, Reg. No. 32,992; Eric S. Hyman, Reg. No. 30,139; Dag H. Johansen, Reg. No. 36,172; William W. Kidd, Reg. No. 31,772; Michael J. Mallie, Reg. No. 36,591; Andre L. Marais, under 37 C.F.R. § 10.9(b); Paul A. Mendonsa, Reg. No. 42,879; Darren J. Milliken, Reg. 42,004; Lisa A. Norris, Reg. No. P44,976; Thien T. Nguyen, Reg. No. 43,835; Thinh V. Nguyen, Reg. No. 42,034; Dennis A. Nicholls, Reg. No. 42,036; Kimberley G. Nobles, Reg. No. 38,255; Daniel E. Ovanezian, Reg. No. 41,236; Babak Redjaian, Reg. No. 42,096; James H. Salter, Reg. No. 35,668; William W. Schaal, Reg. No. 39,018; James C. Scheller, Reg. No. 31,195; Anand Sethuraman, Reg. No. P43,351; Charles E. Shemwell, Reg. No. 40,171; Jeffrey Sam Smith, Reg. No. 39,377; Maria McCormack Sobrino, Reg. No. 31,639; Stanley W. Sokoloff, Reg. No. 25,128; Judith A. Szepesi, Reg. No. 39,393; Vincent P. Tassinari, Reg. No. 42,179; Edwin H. Taylor, Reg. No. 25,129; George G. C. Tseng, Reg. No. 41,355; Joseph A. Twarowski, Reg. No. 42,191; Lester J. Vincent, Reg. No. 31,460; Glenn E. Von Tersch, Reg. No. 41,364; John Patrick Ward, Reg. No. 40,216; Stephen Warhola, Reg. No. 43,237; Charles T. J. Weigell, Reg. No. 43,398; Kirk D. Williams, Reg. No. 42,229; James M. Wu, Reg. No. P45,241; Steven D. Yates, Reg. No. 42,242; Ben J. Yorks, Reg. No. 33,609; and Norman Zafman, Reg. No. 26,250; my patent attorneys, and James A. Henry, Reg. No. 41,064; my patent agent, of BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, with offices located at 12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025, telephone (310) 207-3800, and Alan K. Aldous, Reg. No. 31,905; Robert D. Anderson, Reg. No. 33,826; Joseph R. Bond, Reg. No. 36,458; Richard C. Calderwood, Reg. No. 35,468; Jeffrey S. Draeger, Reg. No. 41,000; Cynthia Thomas Faatz, Reg No. 39,973; Sean Fitzgerald, Reg. No. 32,027; Seth Z. Kalson, Reg. No. 40,670; David J. Kaplan, Reg. No. 41,105; Charles A. Mirho, Reg. No. 41,199; Leo V. Novakoski, Reg. No. 37,198; Naomi Obinata, Reg. No. 39,320; Thomas C. Reynolds, Reg. No. 32,488; Mark Seeley, Reg. No. 32,299; Steven P. Skabrat, Reg. No. 36,279; Howard A. Skaist, Reg. No. 36,008; Steven C. Stewart, Reg. No. 33,555; Raymond J. Werner, Reg. No. 34,752; and Charles K. Young, Reg. No. 39,435; my patent attorneys, and Thomas Raleigh Lane, Reg. No. 42,781; Calvin E. Wells, Reg. No. P43,256; and Peter Lam, Reg. No. P44,855; my patent agents, of INTEL CORPORATION; and James R. Thein, Reg. No. 31,710, my patent attorney; with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

APPENDIX B

Title 37, Code of Federal Regulations, Section 1.56
Duty to Disclose Information Material to Patentability

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclosure information exists with respect to each pending claim until the claim is cancelled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is cancelled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclosure all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) Prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) The closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made or record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application;

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.